On Computational and Statistical Tradeoffs in Matrix Completion with Graph Information

Gautam Dasarathy[a,*], Nikhil Rao[b,*], Richard Baraniuk[a]
[a]Electrical and Computer Engineering, Rice University, Houston, TX, {gautamd,richb}@rice.edu
[b]Technicolor Research, Los Altos, CA, nikhilrao86@gmail.com

I. Introduction

Factoring and completing a partially observed low rank matrix is a widely used technique in collaborative filtering [1], link prediction [2] and sensor network localization [3], among other problems. Scalable methods [4] and rigorous statistical consistency guarantees [5] have been developed for this problem [4].

Most modern datasets however have additional information, either as features or as pairwise relationships between variables. For example, in the case of recommender systems, one can have demographic information or a social network for users. In sensor networks, one might have pairwise similarity information based on the actual locations of sensors. It makes sense to assume that using this additional information will aid in making predictions, and recently, several methods have been proposed to do the same [6], [7]. In this work, we assume that we have access to *weighted graphs* that encode relationships between the variables.

However, the statistical advantage of taking this extra information into consideration comes at a cost. In particular, the computational time of the algorithms developed in [6], [8] depends critically on the number of edges in the graph. While one can obtain speedups trivially by ignoring several edges (or the entire graph), this results in losing (often substantial) statistical advantages obtained via making use of the graph. A natural question then arises:

Can one retain most of the statistical advantages obtained due to the graph information, while still ensuring computational efficiency?

In this work, we take first steps towards answering the above question in the affirmative. We use methods developed for spectral sparsification of graphs [9] to obtain sparse approximations of the graphs in question that retain crucial spectral properties. When the underlying graph is dense, the computational gains of using these approximations can be significant. At the same time, we show that the statistical performance is comparable to using the full graph. In fact, our approach is a first step towards rigorously establishing a graceful and efficient tradeoff between the computational gains and the statistical power of using side information in matrix completion and related problems.

II. Problem Setup, Our Method, and Theory

Suppose $Y \in \mathbb{R}^{m\times n}$ is the data matrix, and we only observe a subset of the entries $Y_{ij}, \forall (i,j) \in \Omega, |\Omega| = N \ll mn$. Furthermore, we assume we have access to the graphs $G_w \in \mathbb{R}^{m\times m}$ and $G_h \in \mathbb{R}^{n\times n}$ which encode the relationships between variables corresponding to the rows and columns of Y respectively. The goal is to estimate the rest of Y. Let L_w, L_h be the corresponding combinatorial graph Laplacians [10]. Then, the problem of matrix completion with graph information can be written as follows:

$$\min_{W,H} \frac{1}{2}\|\mathcal{P}_\Omega(Y - WH^T)\|_F^2 + \frac{\lambda}{2}\{\mathrm{Tr}(W^T L_w W) + \mathrm{Tr}(H^T L_h H)\} \quad (1)$$

where $W \in \mathbb{R}^{m\times k}$, $H \in \mathbb{R}^{n\times k}$, and k is a bound on the rank of Y. $\mathcal{P}_\Omega(\cdot)$ retains those entries of the matrix that lie in the set Ω. Using the graphs yields significant statistical advantages when it comes to estimating Y. At the same time, the natural alternating minimization procedure to solve (1) has a computational cost of $\mathcal{O}\left((N + \mathrm{nnz}(L_h) + \mathrm{nnz}(L_w))\, k\right)$ [6].

Our approach instead is to use sparse approximations of L_h, L_w (say $\widetilde{L}_h, \widetilde{L}_w$) to speed up the algorithm. To this end, we employ spectral sparsification methods of [9], [11]. Let $\widehat{H}$ denote the optimizer of (1) with a W fixed such that $\sigma_{\min}(W) \geq \sigma_w$. Let $\widetilde{H}$ denote the optimizer of (1) with (a) the same W, and (b) $2\widetilde{L}_h$ substituted instead of L_h. Then, we show the following.

Theorem 1: Suppose that $\widetilde{L}_h$ is an $\varepsilon-$close spectral approximation of L_h, then $\widetilde{H}$ satisfies $\|\widetilde{H} - \widehat{H}\|_F^2 \leq \frac{2\lambda(1+\varepsilon)mn}{N\sigma_w^2}\mathrm{Tr}(\widehat{H}^T L_h \widehat{H})$. Furthermore, computational complexity of the entire procedure behaves as $\mathcal{O}(Nk + \frac{n\log n + m\log n}{\varepsilon^2}k)$.

We show a similar statement for optimization w.r.t W. Theorem 1 demonstrates the tradeoff between error in each step of the alt-min procedure and the computational complexity of the procedure. Our initial experiments on both toy and real data suggest that such a graceful computational-statisical tradeoff does exist in large scale matrix factorization problems with graph side information.

III. Experiments

We first tested the methods on toy data. We generated data with power law graphs in the same spirit as [6], with the underlying target matrix of size $m = n = 3K$, rank 20. For all methods considered, we varied $\lambda \in \{10^{-3}, 10^{-2}, \cdots, 10^{1}\}$. We also varied ϵ for graph sparsification, and the number of measurements N obtained for the training set, and corrupted the measurements with AWGN $\sigma = 0.2$. We compute time taken for each method, and the resulting RMSE on the test set (everything in the target matrix not in the training set).

Fig 1 shows that ignoring the graph yields poor performance, as expected from a statistical standpoint. Importantly, as ϵ increases, the performance deteriorates. This is again expected since larger ϵ corresponds to a poorer graph approximation. Note however that unlike the case with no graph, the RMSE is comparable to that obtained using the full graph, even for low number of measurements. Crucially, the time taken (Fig 2) is comparable to that with using no graph, even for small ϵ (0.26). Figures 1 and 2 together show that the statistical performance takes a minor hit, while the computational gains obtained are significant.

We also tested our method on the Epinions dataset [1], which is a recommender system with an accompanying social network among users. We retained the top 8K users from the dataset, corresponding to those with the most connections in the social network. Table I again shows that we incur a very small penalty in the test RMSE, while gaining in terms of time.

*Authors contributed equally

[1] http://www.trustlet.org/epinions.html

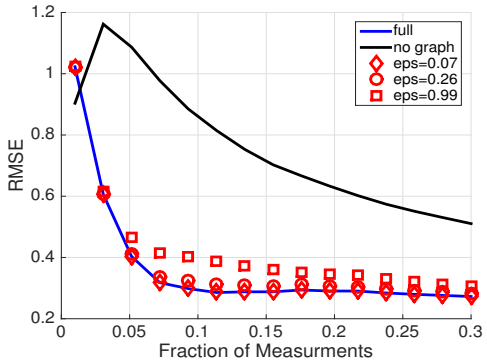

Fig. 1. Test RMSE as number of measurements is varied, p = -0.1

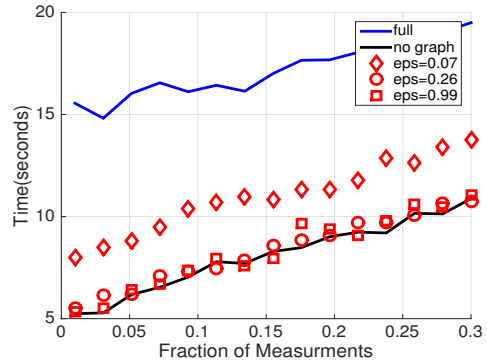

Fig. 2. Time taken as number of measurements is varied, p = -0.1

ϵ	**RMSE**	**time**
full	1.0613	11.186
0.0612	1.0613	9.624
0.0834	1.0614	10.24
0.1138	1.0614	10.296
0.1552	1.0613	10.576
0.2117	1.0614	10.088
0.2887	1.0617	9.297
0.3938	1.0617	9.220
0.5371	1.0620	8.283
0.7325	1.0620	8.867
0.9990	1.0629	8.867

TABLE I

PERFORMANCE ON THE EPINIONS DATASET. AGAIN, THE STATISTICAL DEGRADATION IS NEGLIGIBLE, BUT COMPUTATIONAL GAINS ARE SUBSTANTIAL.

REFERENCES

[1] Y. Koren, R. Bell, C. Volinsky, *et al.*, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[2] A. K. Menon and C. Elkan, "Link prediction via matrix factorization," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 437–452, Springer, 2011.

[3] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pp. 46–54, ACM, 2004.

[4] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis, "Large-scale matrix factorization with distributed stochastic gradient descent," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 69–77, ACM, 2011.

[5] E. J. Candes and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, 2010.

[6] N. Rao, H.-F. Yu, P. K. Ravikumar, and I. S. Dhillon, "Collaborative filtering with graph information: Consistency and scalable methods," in *Advances in Neural Information Processing Systems*, pp. 2107–2115, 2015.

[7] M. Xu, R. Jin, and Z.-H. Zhou, "Speedup matrix completion with side information: Application to multi-label learning," in *Advances in Neural Information Processing Systems*, pp. 2301–2309, 2013.

[8] T. Zhou, H. Shan, A. Banerjee, and G. Sapiro, "Kernelized probabilistic matrix factorization: Exploiting graphs and side information.," in *SDM*, vol. 12, pp. 403–414, SIAM, 2012.

[9] D. A. Spielman and N. Srivastava, "Graph sparsification by effective resistances," *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1913–1926, 2011.

[10] F. R. K. Chung, *Spectral Graph Theory*. Providence, Rhode Island: American Mathematical Society, Dec. 1996.

[11] I. Koutis, A. Levin, and R. Peng, "Improved spectral sparsification and numerical algorithms for sdd matrices," in *STACS'12 (29th Symposium on Theoretical Aspects of Computer Science)*, vol. 14, pp. 266–277, LIPIcs, 2012.